ADAPTIVE COMMUNICATIONS AND SIGNAL PROCESSING LABORATORY
CORNELL UNIVERSITY, ITHACA, NY 14853

A Dynamic Queue Protocol for Multiaccess Wireless Networks with Multipacket Reception

Q. Zhao and L. Tong

Technical Report No. ACSP-TR-11-02-02

November 2002

Abstract

A dynamic medium access control (MAC) protocol is proposed for a finite-user slotted channel with multipacket reception (MPR). This protocol divides the time axis into transmission periods (TPs) where the ith TP is dedicated to the transmission of the packets generated in the $(i-1)$th TP. At the beginning of each TP, the state (active or idle) of each user is estimated based on the length of the previous TP and the incoming traffic load. By exploiting the information on the state of users and the channel MPR capability, the number of users who can simultaneously access the channel in the current TP is chosen so that the expected length of this TP is minimized. As a result, the MPR capability is more efficiently utilized by the proposed protocol as compared to, for example, the slotted ALOHA with optimal retransmission probability. Furthermore, the proposed protocol requires little on-line computation. Its simplicity is comparable to that of slotted ALOHA. It can be applied to random access networks with spread spectrum and/or antenna array†.

Keywords

Medium Access Control. Random Access Network. Multipacket Reception.

I. Introduction

In multiaccess wireless networks where a common channel is shared by a population of users, both the reception capability of the common wireless channel and the efficiency of the medium access control (MAC) protocol affect the network performance. The conventional assumption on the reception capability of the common channel is that a packet is successfully received if and only if there is no concurrent transmissions. Based on such a noiseless collision channel model, MAC protocols are sought to coordinate the transmissions of all users for the efficient utilization of the limited channel resource. Numerous protocols, such as ALOHA [1, 24], the tree algorithm [6], the first-come first-serve (FCFS) algorithm [9], and a class of adaptive schemes [5, 13, 14, 17], have been proposed and their performance studied.

The development of spread spectrum multiple access, space-time coding, and new signal processing techniques makes the correct reception of one or more packets in the presence of other simultaneous transmissions possible. While promising improvement in the overall performance of the network, this multipacket reception (MPR) capability also raises important questions: (1) how does the MPR capability at the physical layer affect the performance of existing MAC protocols? (2) how should we design the MAC layer to fully exploit the MPR capability at the physical layer? Many researchers have provided

†This work was supported in part by the Multidisciplinary University Research Initiative (MURI) under the Office of Naval Research Contract N00014-00-1-0564 and the Army Research Office under Grant ARO-DAAB19-00-1-0507.

answers to the first question. Being the first random access protocol, the application of ALOHA to networks with MPR capability has been thoroughly studied. In [2,8,12,20,27,30,31] and references therein, slotted ALOHA is applied to networks with capture effect. In [10,11], a general model for channels with MPR capability is developed and the performance of slotted ALOHA analyzed for infinite population case. Other random access protocols such as the FCFS algorithm and the window protocol [22] have also been extended to networks with capture effect and their performance evaluated [3,19,26,28]. The application of contention free scheme TDMA to networks with MPR capability is another interesting research topic. In [7,16], the authors address the use of TDMA in fully connected half-duplex ad hoc networks with MPR provided by multiple independent collision channels. In [25], dynamic time slot allocation is introduced for cellular systems with antenna arrays. Given a set of active users (users with packets to transmit), the proposed dynamic slot allocation scheme assigns an appropriate number of active users to each time slot to utilize the MPR capability provided by the antenna array.

Answers to the second question, however, are scarce in the literature. The Multi-Queue Service Room (MQSR) protocol proposed in [29] is perhaps the first MAC protocol designed explicitly for networks with MPR capability. By optimally exploiting all available information up to the current slot, this protocol grants access to the channel to an appropriate subset of users so that the expected number of successfully received packets is maximized in each slot, leading to the optimal utilization of the channel MPR capability. The difficulty of the MQSR protocol, however, lies in its computational complexity which grows exponentially with the number of users in the network.

In this paper, we propose, for general MPR channels, a MAC protocol that achieves a performance comparable to that of the MQSR protocol, but with a much simpler implementation. Similar to the structure of collision resolution interval in the dynamic tree protocol [5], the proposed scheme, referred to as the dynamic queue protocol, divides the time axis into transmission periods (TP) where the ith TP is dedicated to the transmission of the packets generated in the $(i-1)$th TP. With such a transmission structure, our knowledge on the state of users at the beginning of the ith TP can be characterized by the probability q_i that a user has a packet to transmit in the ith TP, which depends on the incoming traffic load and the duration of the $(i-1)$th TP. Based on q_i and the channel MPR capability, the size of the access set which contains users who can access the channel simultaneously is chosen for the ith TP so that the expected duration of this TP is minimized, *i.e.,* all packets generated in the $(i-1)$th TP are successfully transmitted within a minimum number of slots. As a consequence, unnecessary empty slots at light traffic and excessive collision events at heavy traffic are avoided simultaneously, leading to an efficient utilization of the channel MPR capability at any incoming traffic load. Furthermore, the optimal size of access set for each TP is obtained from a look-up table. The on-line implementation of the proposed protocol is as simple as that of slotted ALOHA.

This paper is organized as follows. In Section II, we present the model of a communication network with MPR capability. In Section III, we propose the dynamic queue protocol and draw connections between it and existing protocols such as the dynamic tree protocol [5] and the MQSR protocol [29]. An analysis on the throughput and delay performance of the dynamic queue protocol is presented in Section IV. Finally, numerical and simulation examples are presented in Section V, where we compare the throughput and average delay of the proposed dynamic queue protocol with that of the optimal scheme MQSR [29] and the slotted ALOHA with optimal retransmission probability.

II. The Model

We consider a communication network with M users who transmit data to a central controller through a common wireless channel. Each user generates data in the form of equal-sized packets. Transmission time is slotted, and each packet requires one time slot to transmit. With probability p, a user independently generates a packet within each slot.

The common wireless channel is characterized by the probability of having k successes in a slot when there are n transmissions as denoted by

$$C_{n,k} = \mathsf{P}[k \text{ packets are correctly received} \mid n \text{ are transmitted}] \; (1 \leq n \leq M, 0 \leq k \leq n).$$

The multipacket reception matrix of the channel in a network with M users is then defined as

$$\mathbf{C} = \begin{pmatrix} C_{1,0} & C_{1,1} & & & \\ C_{2,0} & C_{2,1} & C_{2,2} & & \\ \vdots & \vdots & \vdots & & \\ C_{M,0} & C_{M,1} & C_{M,2} & \cdots & C_{M,M} \end{pmatrix}. \tag{1}$$

By choosing $C_{1,0} > 0$, this channel model can easily accommodate a noisy environment. Let

$$\mathcal{C}_n \triangleq \sum_{k=1}^{n} k C_{n,k} \tag{2}$$

denote the expected number of correctly received packets when total n packets are transmitted. We then define the capacity of an MPR channel as

$$\eta \triangleq \max_{n=1,\cdots,M} \mathcal{C}_n. \tag{3}$$

Note that the channel capacity we define here differs from that defined by Shannon in information theory. As defined in (3), η is the maximum number of packets that we can expect to successfully receive within one time slot. It is the maximum throughput the MPR channel can offer, independent of MAC protocols. Let

$$n_0 \triangleq \min\{\arg \max_{n=1,\cdots,M} \mathcal{C}_n\}. \tag{4}$$

We can see that at heavy traffic load, n_0 packets should be transmitted simultaneously to achieve the channel capacity η. Noticing that the number of simultaneously transmitted packets for achieving η may not be unique, we define n_0 as the minimum to save transmission power. For MPR channels with n_0 greater than 1, contention should be preferred at any traffic load in order to fully exploit the channel MPR capability.

This general model for MPR channels, also considered in [4,10,11], applies to the conventional collision channel and channels with capture as special examples. The reception matrix of the conventional noiseless collision channel and channels with capture are given by

$$\begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 0 & 0 & \cdots & 0 \end{pmatrix}, \qquad \begin{pmatrix} 1-p_1 & p_1 & 0 & \cdots & 0 \\ 1-p_2 & p_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 1-p_M & p_M & 0 & \cdots & 0 \end{pmatrix}, \tag{5}$$

where p_i is the probability of capture given i simultaneous transmissions.

Access to the common wireless channel is controlled by the central controller. At the beginning of each slot, the central controller chooses and broadcasts an access set which contains users allowed to access the channel in this particular slot. Users and only users in this access set transmit packets if they have any. At the end of this slot, the central controller observes the channel outcome which contains information on whether this slot is empty and whose packets are successfully received. Here we assume that the central controller can distinguish without error between empty and nonempty slots. Furthermore, if some packets are successfully demodulated at the end of a slot, the central controller can identify the source of these packets. However, if at least one packet is successfully demodulated at the end of a slot, the central controller does not assume the knowledge whether there are other packets transmitted in this slot but not successfully received. We illustrate this point in Figure 1 where we consider possible outcomes of a slot: empty, nonempty with success, and nonempty without success (successfully received packets are illustrated by shaded rectangles). To the central controller, the two events that happened in the third and the forth slot are indistinguishable.

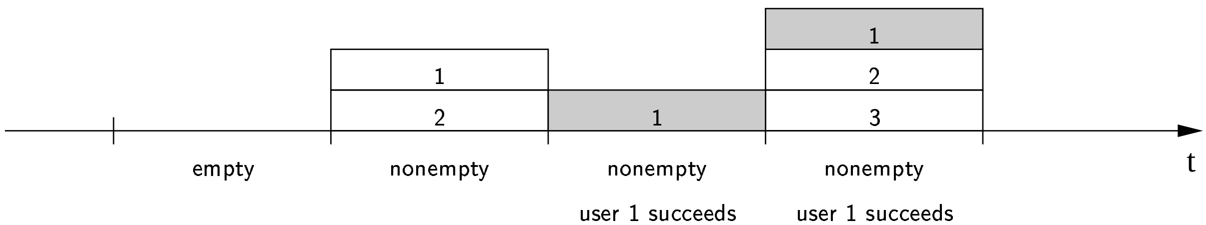

Fig. 1: Possible outcomes of a slot

figure

After observing the channel outcome, the central controller acknowledges the sources of successfully received packets (if any). Users who transmit but do not receive acknowledgment assume their packets are lost and will retransmit the next time they are enabled. We assume in this paper that the down link channel (from the central controller to the users) is error free and the time for acknowledgment and broadcasting the access set is negligible.

Our goal here is to design, for a multiaccess network as specified above, a MAC protocol that adaptively controls the number of users simultaneously accessing the channel according to the channel MPR capability and the current traffic load. It should achieve efficient channel utilization with a simple on-line implementation.

III. The Dynamic Queue Protocol

A. The Structure of Transmission Period

In the proposed dynamic queue protocol, the time axis is divided into transmission periods (TPs) as illustrated in Figure 2, where we assume that the network starts at time 0 and one slot lasts one time unit. Each TP is dedicated to the transmission of packets generated in the previous TP and ends when the central controller can assert that all packets generated in the previous TP have been successfully transmitted.

We assume that besides the packet waiting for transmission in the current TP, each user can hold at most one packet newly generated in the current TP and to be transmitted in the next one. Thus, in

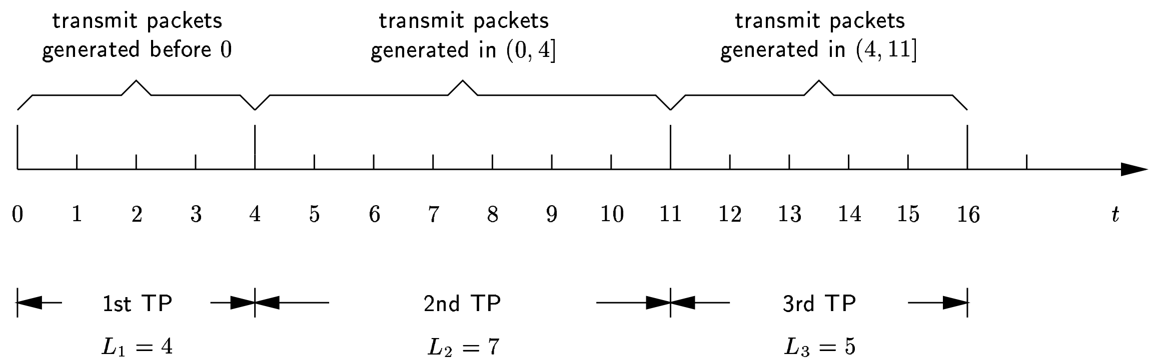

Fig. 2: The structure of transmission period.

figure

each TP, each user has at most one packet to transmit. Let q_i denote the probability that a user has a packet to transmit in the ith $(i \geq 1)$ TP. Recall that p denotes the probability that a user generates a packet within one time slot. We have

$$q_i = 1 - (1-p)^{L_{i-1}}, \tag{6}$$

where L_{i-1} $(i \geq 2)$ denotes the length of the $(i-1)$th TP defined as the number of slots it contains; L_0 specifies the network initial condition and is known to the central controller. Thus, q_i carries our knowledge on the state of each user at the beginning of the ith TP. Based on q_i and the channel reception matrix $\mathbf{C}$, the size N_i of the access set which contains users who can simultaneously access the channel in the ith TP is chosen optimally (see Section III-C). Packets generated in the $(i-1)$th TP are then transmitted according to the procedure specified in Section III-B.

B. The Structure of the Dynamic Queue Protocol

The basic structure of the dynamic queue protocol is a waiting queue. At the beginning of the ith TP, all M users are waiting in a queue for the transmission of their packets generated in the $(i-1)$th TP. A user is said to be processed if the central controller detects either it does not generate packet in the $(i-1)$th TP or its packet generated in the $(i-1)$th TP has been successfully transmitted. Based on q_i given by (6), N_i, the size of the access set for this TP, is chosen. Then, the first N_i users in the queue are enabled to access the channel in the first slot of the ith TP. At the end of this slot, the central controller detects whether this slot is empty or not. If it is empty, all these N_i users are processed and the next N_i users in the queue are enabled in the next slot. On the other hand, if this slot is not empty and k ($k \geq 0$) packets are successfully received, the sources of these k packets are processed and removed from the waiting queue; the rest $N_i - k$ users along with the next k users in the queue are enabled to access the channel in the next slot. This procedure continues until all M users are processed.

We illustrate in Figure 3 the basic procedure of the dynamic queue protocol, where we consider the ith TP in a network with $M = 6$. Suppose that each of user 2, 5, and 6 (shaded with dashed lines) has generated a packet in the $(i-1)$th TP, and we choose $N_i = 3$ for the ith TP. As shown in Figure 3, at the beginning of the ith TP, all 6 users are waiting in a queue to access the channel. In the first slot of this TP, user 1, 2, and 3 are enabled, resulting in a successful transmission by user 2 (shaded rectangles indicate successfully transmitted packets). User 2 is then processed and removed from the waiting queue. User 1, 3 and 4 are then enabled in the second slot of this TP, resulting in an empty slot in which all these three users are processed. Finally, the only two users left in the queue, namely, user 5 and 6, access the channel in the third slot. At the end of the fourth slot, all users are processed, leading to the end of the ith TP and the beginning of the $(i+1)$th TP.

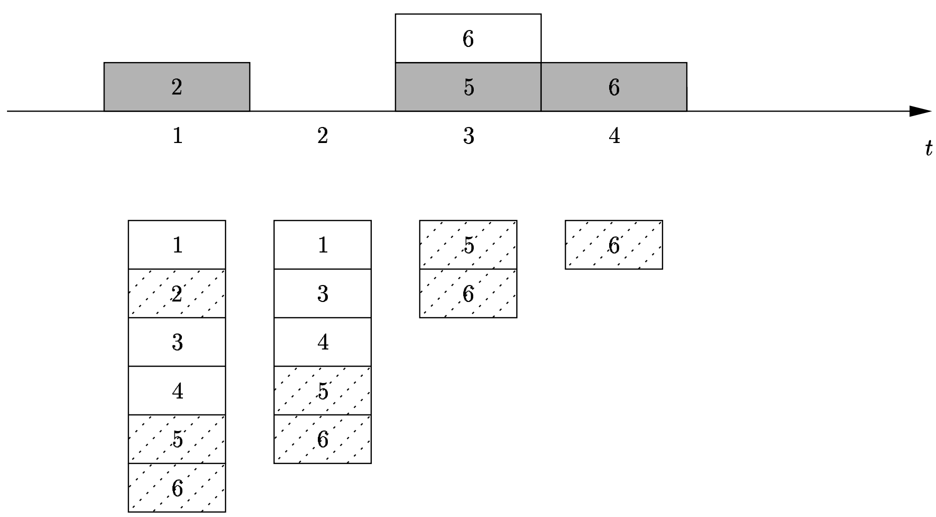

Fig. 3: The basic procedure of the dynamic queue protocol.

figure

With this structure, the only parameter to be designed is N_i, the size of the access set for the ith ($i \geq 1$) TP, which we discuss in Section III-C.

We point out that the order of a user in the waiting queue affects its average packet delay. While all users generate packets simultaneously, users in the front of the queue access the channel before users in the end of the queue unless $N_i = M$. If priority among users is desirable, users with higher priority should be in the front of the queue in each TP. Otherwise, the order of a user in the queue needs to be randomized at the beginning of each TP to ensure fairness. In Section V, simulation examples show that the average delay for the last user in the queue can be twice as large as that for the first user at medium and heavy traffic load.

C. The Optimal Access Set

The optimal size N_i of the access set for the ith TP is chosen so that the expected length of this TP is minimized, *i.e.*, the expected number of slots for processing all M users, each with probability q_i having a packet, is minimized. Specifically, N_i is determined by

$$N_i = \arg \min_{N=1,\cdots,M} E[L_i \mid q_i, N], \tag{7}$$

where $E[L_i \mid q_i, N]$ is the expected length of the ith TP when each user with probability q_i has a packet to transmit and the size of the access set is N.

In order to determine N_i, we calculate $E[L_i \mid q_i, N]$ as the absorbing time of a finite state discrete Markov chain. It can be shown that the number of unprocessed users at the beginning of a slot along with the number of packets that will be transmitted in this slot forms a Markov chain. Specifically, at the beginning of a slot in the ith TP, the network is in state (j, k) if there are j $(j = 0, \cdots, M)$ unprocessed users and k $(k = 0, \cdots, \min\{N, j\})$ packets to be transmitted in this slot when the size of access set is chosen to be N. A state diagram of this Markov chain for $M = 2$ and $N = 1$ is illustrated in Figure 4. With probability q_i, the first user in the queue has a packet to transmit in the ith TP. Thus, with probability q_i, the Markov chain starts with state $(2, 1)$, and with probability $1 - q_i$, it starts with state $(2, 0)$. Take the state $(2, 1)$ for example. With probability $C_{1,0}$, the transmission by the first user in the queue does not succeed. The chain then stays in $(2, 1)$. With probability $C_{1,1}q_i$, the transmission by the first user succeeds and the second user in the queue has a packet. The chain then jumps to state $(1, 1)$. With probability $C_{1,1}(1 - q_i)$, the chain jumps to state $(1, 0)$.

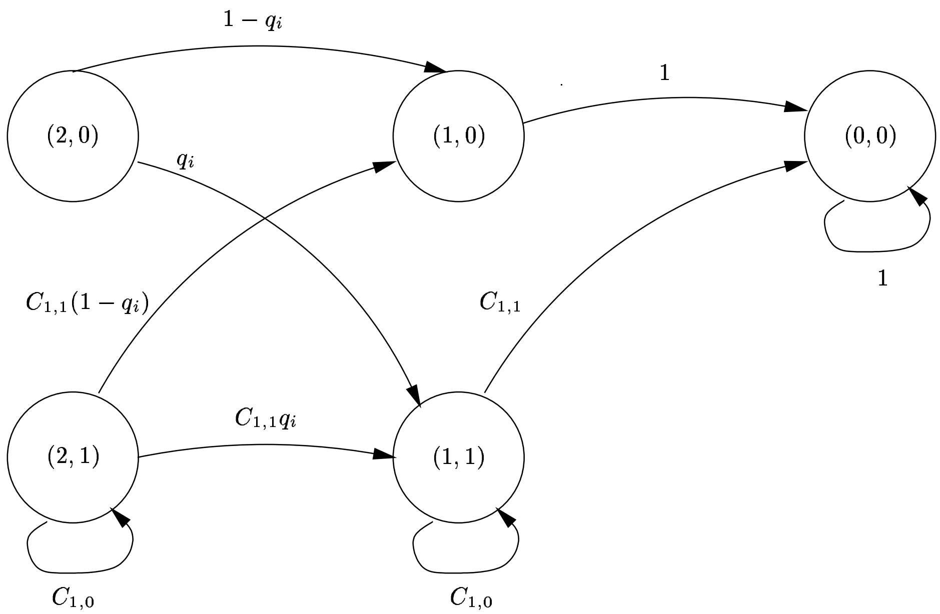

Fig. 4: A state transition diagram.

figure

In general, the transition probability from state (j,k) to state (l,m) is given by

$$r_{(j,k),(l,m)} = \begin{cases} B(m, \min\{N,l\}, q_i) \\ (\text{if } k=0, l=\max\{j-N,0\}, 0 \le m \le \min\{N,l\}) \\ C_{k,j-l} B(m-k+j-l, \min\{j-l, \max\{j-N,0\}\}, q_i) \\ (\text{if } 1 \le k \le \min\{N,j\}, j-k \le l \le j, k-(j-l) \le m \le \min\{k,l\}) \\ 0 \quad (\text{otherwise}) \end{cases}, \tag{8}$$

where $B(u,U,s)$ denote the probability mass at the value u of a Binomial random variable with total U trials and a success probability s, *i.e.*,

$$B(u,U,s) \triangleq \binom{U}{u} s^u (1-s)^{U-u}. \tag{9}$$

The initial condition of this Markov chain is given by

$$P[X_0 = (M,k)] = B(k,N,q_i), \quad k = 0, \cdots, N, \tag{10}$$

where X_0 denote the initial state of the Markov chain. With state (0,0) defined as the absorbing state, $E[L_i \mid q_i, N]$ is the absorbing time of this Markov chain, which is defined as the expected number of transitions until the first hit of state (0,0). Define $e_{(j,k)}$ as the expected remaining time until absorption given that the current state is (j,k). Let

$$\mathbf{e} \triangleq [e_{(M,0)}, \cdots, e_{(M,N)}, e_{(M-1,0)}, \cdots, e_{(1,0)}, e_{(1,1)}, e_{(0,0)}]^t. \tag{11}$$

We then have

$$(\mathbf{I} - \mathbf{P})\mathbf{e} = \mathbf{1}, \tag{12}$$

where $\mathbf{P}$ is the transition probability matrix with entries specified by (8), $\mathbf{I}$ and $\mathbf{1}$ denote, respectively, an identity matrix and a vector with all entries equal to 1.

¿From (12), we can solve for $e_{(M,k)}$ for $k = 0, \cdots, N$. Thus, considering the initial condition of the Markov chain given by (10), we can calculate $E[L_i \mid q_i, N]$ as

$$E[L_i \mid q_i, N] = \sum_{k=0}^{N} B(k, N, q_i) e_{(M,k)}. \tag{13}$$

With $E[L_i \mid q_i, N]$ computed for all possible N, the optimal size N_i of the access set for the ith TP can be easily obtained from (7).

We point out that the optimal size of the access set can be computed off line. By varying q_i from 0 to 1, we can construct a table that specifies the interval of q_i in which a size $N \in \{1, \cdots, M\}$ of the access set is optimal (a typical look-up table is illustrated in Figure 8). Thus, when the network starts, the optimal size of the access set for each TP can be obtained from this table; little on-line computation is required to implement the dynamic queue protocol.

The basic procedure of the dynamic queue protocol is given in Figure 5.

D. Connections with Existing MAC Protocols

In this section, we draw connections between the proposed dynamic queue protocol and existing MAC schemes. In particular, we consider the dynamic tree [5] protocol proposed for the noiseless collision channel and the MQSR protocol [29] proposed for MPR channels.

D.1 The Dynamic Tree Protocol

The similarity between the dynamic queue and the dynamic tree protocol is the structure of transmission period. Although the terminology of collision resolution interval (CRI) is used in the dynamic tree protocol, both protocols have the property that newly generated packets can not be transmitted

The Dynamic Queue Protocol

In the ith ($i \geq 1$) TP,

1. Let W denote the number of unprocessed users. Set $W = M$.
2. Compute q_i as given by (6).
3. Choose N_i based on q_i from the look-up table.
4. The first N_i users form the access set.
5. Users in the access set access the channel in the current slot.
6. At the end of this slot,

- if the slot is empty, update W as $W = W - A$, where A is the size of the access set; remove all users from the access set; the next $\min\{N_i, W\}$ users in the queue form the access set;
- if the slot is not empty and k packets are successfully received, remove the source of these k packets from the access set; update W as $W = W - k$; the next $\min\{k, W\}$ users in the queue join the access set.

7. Repeat Step 5 and Step 6 until $W = 0$. This starts the $(i + 1)$th TP.

Fig. 5: The dynamic queue protocol.

figure

until the current TP (CRI) ends. This ensures that a single parameter q_i is sufficient to characterize our knowledge on the state of users at the beginning of each TP (CRI).

The main difference between these two protocols lies in their schemes of determining the access set for each slot in a particular TP (CRI). Proposed exclusively for the noiseless collision channel, the dynamic tree protocol utilizes the binary tree algorithm [6] for determining the access set for each slot. However, like other splitting algorithms such as FCFS [9], the tree algorithm relies on three assumptions that do not hold in a general MPR channel. First, the tree algorithm relies on the property that a successful transmission in the noiseless collision channel implies that other users in the access set do not have packets. Thus, in a slot with a success of one user, all users in the access set are processed. This, however, is not true in a general MPR channel (see Figure 1) where any user in the access set from whom we do not receive a packet in a nonempty slot is unprocessed. If we insist on the tree structure in protocols designed for MPR channels, the access set may have to be unnecessarily shrunk in order to enable the unprocessed users left in the access set after a slot with success. This is due to the boundaries among nodes on the same level of a tree. The second assumption made by the dynamic tree protocol is that packet collisions can only be resolved by splitting of users. However, the MPR capability opens new options for resolving a collision. Splitting is not always necessary or even sensible. Take, for example, an MPR channel with $C_{2,0} = \epsilon$ and $C_{2,2} = 1 - \epsilon$ where $0 < \epsilon \ll 1$. When two packets are simultaneously transmitted and none successfully received, in stead of splitting, it may be more sensible to enable the same set of users again. Finally, the dynamic tree protocol does not take noise into account. Any nonempty slot with no success is assumed to be a consequence of collision, hence leads to a splitting of users. This causes unnecessary empty slots in the case that only one user in the access set transmits but does not succeed because of noise.

In the dynamic queue protocol, in stead of a tree, a queue structure is utilized for determining the access set for each slot. The boundaries among users are eliminated by the queue structure. By letting the next k users in the waiting queue join the access set after a slot with k successes, we can keep the size of the access set to be N_i which has been chosen optimally (in terms of minimizing $E[L_i]$). Furthermore, the same N_i users are enabled after a nonempty slot with no success, where N_i has been chosen according to the channel MPR capability. This enables us to exploit the MPR capability for collision resolution (in the case of more than one active user) and avoid unnecessary splitting of users (in the case of one active user in a noisy environment). Note that $E[L_i \mid q_i, N] = \infty$ if $\mathcal{C}_N = 0$. We thus have

$$N_i \in \{N : \ 1 \leq N \leq M, \ \mathcal{C}_N > 0\}, \tag{14}$$

i.e., the optimal size of access set always enables us to resolve packet collisions via the channel MPR capability.

D.2 The Multi-Queue Service Room (MQSR) Protocol

As their names suggest, both the MQSR and the dynamic queue protocol utilizes a queue structure for determining the access set for each slot. The difference between them is the amount of information they exploit for choosing the size of access set.

In the MQSR protocol, in order to maximize per-slot throughput, the size of access set for each slot is chosen by exploiting all the information that is available at the beginning of this slot. Since the outcome of each slot provides information on the state of users, the size of access set is updated at the beginning of each slot in order to incorporating the newly available information. However, this update of the size of access set on a slot-by-slot basis costs the high computational complexity of the MQSR protocol.

With the structure of transmission period, the dynamic queue protocol utilizes only the information available at the beginning of each TP for determining the size of access set. Once the size of the access set is chosen at the beginning of a TP, it is used in all slots in this TP, except when the number of unprocessed users left in the waiting queue is smaller than the size of the access set. Information obtained from the outcome of each slot within a TP is used only for determining whether a user is processed or not, but not for updating the size of the access set. It turns out that by fixing the size of access set for the whole TP, determining it becomes as simple as looking up a table. The price we paid for this simple implementation is performance. Nevertheless, extensive simulations demonstrate that the performance of the dynamic queue protocol is comparable to that of the optimal MQSR protocol (see Figure 9,12).

IV. Steady-State Performance Analysis

Our main concern with MAC protocols is their long term behavior (when the initial condition of the network becomes irrelevant). Thus, steady-state performance measures such as throughput and average delay are commonly used for evaluating a MAC protocol. In this section, we study the steady-state performance of the dynamic queue protocol using throughput and average delay as our measures. First, we show that the network employing the dynamic queue protocol eventually reaches a steady state, regardless of the initial condition L_0. We then derive, in Section IV-B, formulas for throughput and average delay provided by the dynamic queue protocol at an arbitrary traffic load.

A. The Existence of Steady State

Given the channel reception matrix $\mathbf{C}$ and the incoming traffic load p, the optimal size N_i of the access set for the ith TP is a function of L_{i-1}, *i.e.*,

$$N_i = f_{\mathbf{C},p}(L_{i-1}). \tag{15}$$

In general, $f_{\mathbf{C},p}(\cdot)$ is a monotonic decreasing ‡ function as illustrated in Figure 6. It is completely determined by $\mathbf{C}$ and p and can be computed off line. Suppose that the range of $f_{\mathbf{C},p}(\cdot)$ is $\{\underline{n}_1, \cdots, \underline{n}_J\}$ with $\underline{n}_1 < \underline{n}_2 < \cdots < \underline{n}_J$. We then define

$$\underline{l}_j \triangleq \min\{l : f_{\mathbf{C},p}(l) = \underline{n}_j\}, \quad j = 1, \cdots, J. \tag{16}$$

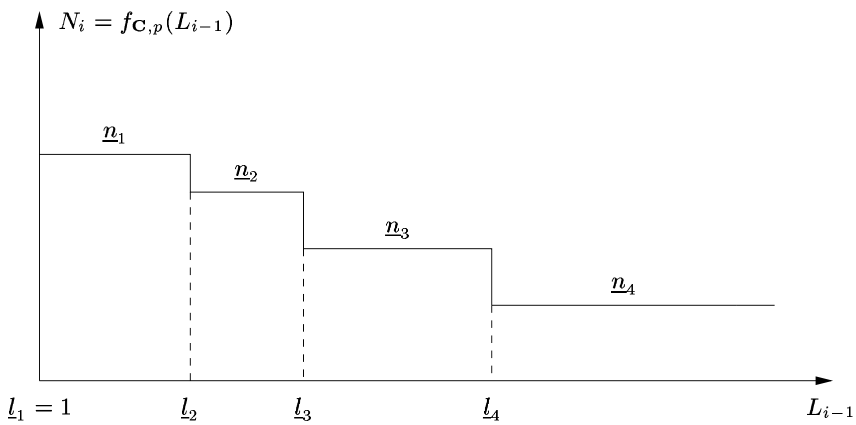

Fig. 6: N_i as a function of L_{i-1}.

figure

It can be shown (see Appendix A) that $\{L_i\}_{i=0}^{\infty}$ is a homogeneous Markov process with infinite state space $\mathcal{S} = \mathcal{Z}^+$ ($\mathcal{Z}^+$ denotes the set of positive integers) and transition probability

$$p_{l,m} \triangleq P[L_i = m \mid L_{i-1} = l]. \tag{17}$$

The steady state of a network using the dynamic queue protocol is then defined as the stationary distribution of $\{L_i\}_{i=0}^{\infty}$. Before using steady-state performance measures such as throughput and average delay, questions about the existence and uniqueness of the network steady states must be resolved.

Theorem 1: Suppose that $f_{\mathbf{C},p}(\cdot)$ is a monotone decreasing function with range $\{\underline{n}_1, \cdots, \underline{n}_J\}$. Let

$$\mathcal{Q}_1 = \{1, 2, \cdots, \lceil \frac{M}{\underline{n}_w} \rceil - 1\}, \quad \mathcal{Q}_2 = \{\lceil \frac{M}{\underline{n}_w} \rceil, \lceil \frac{M}{\underline{n}_w} \rceil + 1, \cdots\} \tag{18}$$

‡A heuristic argument for $f_{\mathbf{C},p}(\cdot)$ being monotonic decreasing is as follows. In order to process all packets generated in the $(i-1)$th TP within a minimum number of slots, n_0 (as defined in (4)) packets should be transmitted simultaneously in each slot. With a smaller L_{i-1}, the probability that a user has a packet to transmit in the ith TP is smaller. Hence, the access set for the ith TP need to be enlarged so that the total number of packets held by users in the access set approaches n_0.

be a partition of the state space $\mathcal{S}$, where

$$w \triangleq \max\{j : \lceil \frac{M}{\underline{n}_j} \rceil \geq \underline{l}_j, 1 \leq j \leq J\}. \tag{19}$$

Consider a noisy environment with $0 < C_{1,0} < 1$. We have, for $p \in (0,1)$,

T1.1 all states in $\mathcal{Q}_1$ are transient;

T1.2 if the initial distribution of $\{L_i\}_{i=0}^{\infty}$ is such that $P[L_0 \in \mathcal{Q}_2] = 1$, then $\{L_i\}_{i=0}^{\infty}$ is ergodic;

T1.3 $\{L_i\}_{i=0}^{\infty}$ has a limiting distribution $\{\pi_l\}_{l \in \mathcal{S}}$ satisfying

$$\pi_l \begin{cases} > 0 & \text{if } l \in \mathcal{Q}_2 \\ = 0 & \text{if } l \in \mathcal{Q}_1 \end{cases} . \tag{20}$$

The proof of Theorem 1 is based on the following properties of $\{L_i\}_{i=0}^{\infty}$ (proved in Appendix B).

Property 1: Let $l \to m$ denote that state l leads to state m. Under the same assumptions in Theorem 1, The Markov process $\{L_i\}_{i=0}^{\infty}$ has the following properties.

P1.1 For $l, m \in \mathcal{S}$ with $m \geq \lceil \frac{M}{\underline{n}_j} \rceil$, $l < \underline{l}_{j+1}$, where $j = 1, \cdots, J$ and $\underline{l}_{J+1} \triangleq \infty$, we have

$$l \to m. \tag{21}$$

P1.2 For $l, m \in \mathcal{S}$ with $m < \lceil \frac{M}{\underline{n}_w} \rceil$, $l \geq \lceil \frac{M}{\underline{n}_w} \rceil$, we have

$$l \not\to m. \tag{22}$$

Proof of Theorem 1:

T1.1: Consider a state $l \in \mathcal{Q}_1$. Since $l < \underline{l}_{J+1} \triangleq \infty$, we have, from *P1.1*,

$$l \to \lceil \frac{M}{\underline{n}_J} \rceil. \tag{23}$$

Since $\underline{n}_w \geq \underline{n}_J$, we have,

$$\lceil \frac{M}{\underline{n}_J} \rceil \geq \lceil \frac{M}{\underline{n}_w} \rceil. \tag{24}$$

Hence, by the fact that $l < \lceil \frac{M}{\underline{n}_w} \rceil$ and *P1.2*,

$$\lceil \frac{M}{\underline{n}_J} \rceil \not\to l. \tag{25}$$

We then conclude from (23,25) that there is a positive probability of the chain leaving state l and never coming back, *i.e.*, state l is transient.

T1.2: Consider a state $l \in \mathcal{Q}_2$, *i.e.*, $l \geq \lceil \frac{M}{\underline{n}_w} \rceil$. By *P1.2*,

$$l \not\to m \tag{26}$$

for any $m \in \mathcal{Q}_1$. Hence, when we restrict the initial state to $\mathcal{Q}_2$, the state space of $\{L_i\}_{i=0}^{\infty}$ becomes $\mathcal{Q}_2$. To show $\{L_i\}_{i=0}^{\infty}$ is ergodic under this initial condition, we need to show it is irreducible, aperiodic, and positive recurrent.

Since $l < \underline{l}_{J+1} \triangleq \infty$ holds for any l, *P1.1* implies that any state leads to a state no smaller than $\lceil \frac{M}{\underline{n}_J} \rceil$. Hence, any two states no smaller than $\lceil \frac{M}{\underline{n}_J} \rceil$ communicate. It then follows that when $w = J$, the chain is irreducible. When $w < J$, we only need to show that for any $\lceil \frac{M}{\underline{n}_w} \rceil \leq m < \lceil \frac{M}{\underline{n}_J} \rceil$, we have m as a consequence of $\lceil \frac{M}{\underline{n}_J} \rceil$. Since $\lceil \frac{M}{\underline{n}_j} \rceil < \underline{l}_j$ for $w < j \leq J$ by the definition of w, we have, from *P1.1*,

$$\lceil \frac{M}{\underline{n}_j} \rceil \to \lceil \frac{M}{\underline{n}_{j-1}} \rceil, \quad w < j \leq J. \tag{27}$$

Thus,

$$\lceil \frac{M}{\underline{n}_J} \rceil \to \lceil \frac{M}{\underline{n}_{J-1}} \rceil \to \cdots \to \lceil \frac{M}{\underline{n}_w} \rceil. \tag{28}$$

Since $\lceil \frac{M}{\underline{n}_w} \rceil \leq \lceil \frac{M}{\underline{n}_{w+1}} \rceil < \underline{l}_{w+1}$, and $m \geq \lceil \frac{M}{\underline{n}_w} \rceil$, we have, by *P1.1*,

$$\lceil \frac{M}{\underline{n}_w} \rceil \to m. \tag{29}$$

This completes the proof of the chain being irreducible. $\{L_i\}_{i=0}^{\infty}$ being aperiodic follows directly from the self loop at state $\lceil \frac{M}{\underline{n}_J} \rceil$.

We now show, through Pakes Lemma [21], that this irreducible and aperiodic Markov chain is ergodic. Let d_l be the drift at state l defined as

$$d_l \triangleq E[L_i - L_{i-1} \mid L_{i-1} = l]. \tag{30}$$

We have, for any $l \in \mathcal{Q}_2$,

$$\begin{aligned} d_l &= E[L_i \mid L_{i-1} = l] - l \\ &\leq \frac{M}{C_{1,1}} - l, \end{aligned} \tag{31}$$

where (31) follows from the fact that

$$E[L_i \mid L_{i-1} = l] \leq E[L_i \mid N_i = 1] = \frac{M}{C_{1,1}}. \tag{32}$$

Thus,

$$\limsup_{l\to\infty} d_l < 0. \tag{33}$$

By Pakes Lemma, we conclude that $\{L_i\}_{i=0}^{\infty}$ is ergodic.

T1.3: This statement follows directly from *T1.1* and *T1.2*.

□□□

Theorem 1 shows that a network which employs the dynamic queue protocol will eventually reach a unique steady state, regardless of the initial condition L_0. Thus, we can use measures such as throughput and average delay to study the long term behavior of the dynamic queue protocol.

B. Throughput and Packet Delay

B.1 Throughput

The throughput U is defined as the average number of packets successfully transmitted within one time slot. Let S_i denote the number of packets generated in the ith TP. Recall that packets generated in the ith TP are all successfully transmitted in the $(i+1)$th TP. We have

$$U = \lim_{i\to\infty} \frac{S_0 + S_1 + \cdots + S_{i-1}}{L_1 + L_2 + \cdots + L_i}. \tag{34}$$

By Theorem 1, $\{L_i\}_{i=0}^{\infty}$ is an ergodic process with limiting distribution $\{\pi_l\}_{l\in\mathcal{S}}$ given by (20). Hence, at steady state, we have, for any $p \in (0,1)$,

$$\begin{aligned} U &= \frac{E[S_i]}{E[L_i]} \\ &= \frac{\sum_{l\in\mathcal{Q}_2}(\sum_{n=0}^{M} nB(n,M,1-(1-p)^l))\pi_l}{\sum_{l\in\mathcal{Q}_2} l\pi_l} \\ &= \frac{\sum_{l\in\mathcal{Q}_2}(1-(1-p)^l)M\pi_l}{\sum_{l\in\mathcal{Q}_2} l\pi_l}. \end{aligned} \tag{35}$$

B.2 Average Packet Delay

The average packet delay D is defined as the average number of slots from the time a packet is generated to that it is successfully transmitted. Since a packet generated in the ith TP is transmitted in the $(i+1)$th TP, the average packet delay is determined by the lengths of two consecutive TPs. Based on the Markovian property of $\{L_i\}_{i=0}^{\infty}$, we can show that $\{(L_i, L_{i+1})\}_{i=0}^{\infty}$ is a homogeneous Markov process with infinite state space $\hat{\mathcal{S}} \in \mathcal{Z}^+ \times \mathcal{Z}^+$. Its transition probability is given by

$$\begin{aligned} \hat{p}_{(j,k),(l,m)} &\stackrel{\Delta}{=} P[(L_i, L_{i+1}) = (l,m) \mid (L_{i-1}, L_i) = (j,k)] \\ &= \begin{cases} 0 & \text{if } l \neq k \\ p_{l,m} & \text{if } l = k \end{cases} . \end{aligned} \tag{36}$$

Furthermore, if $\{L_i\}_{i=0}^{\infty}$ has a limiting distribution $\{\pi_l\}_{l \in \mathcal{S}}$ given by (20), then $\{(L_i, L_{i+1})\}_{i=0}^{\infty}$ has a limiting distribution $\{\hat{\pi}_{(l,m)}\}_{(l,m)\in\hat{\mathcal{S}}}$ given by

$$\hat{\pi}_{(l,m)} = \pi_l p_{l,m} \begin{cases} > 0 & \text{if } l, m \in \mathcal{Q}_2 \\ = 0 & \text{otherwise} \end{cases} , \tag{37}$$

which follows from

$$\hat{\pi}_{(l,m)} \stackrel{\Delta}{=} \lim_{i\to\infty} P[(L_i, L_{i+1}) = (l,m)] = \lim_{i\to\infty} P[L_i = l] p_{l,m} = \pi_l p_{l,m}. \tag{38}$$

Based on the limiting distribution of $\{(L_i, L_{i+1})\}_{i=0}^{\infty}$, the average delay provided by the dynamic queue protocol can be calculated as follows.

In the steady state, with probability $\hat{\pi}_{(l,m)}$, a packet is generated in a TP with length l and successfully transmitted in a TP with length m. Without loss of generality, we assume these two transmission periods are the first and the second TP. Let t_g and t_s denote, respectively, the time instance that the packet is generated and that the packet is successfully received. Assuming that the first TP starts at time 0 and each slot lasts one time unit, we have

$$\begin{aligned} D &= \sum_{l,m\in\mathcal{Q}_2} E[t_s - t_g \mid L_1 = l, L_2 = m]\hat{\pi}_{(l,m)} \\ &= \sum_{l,m\in\mathcal{Q}_2} (E[t_s \mid L_1 = l, L_2 = m] - E[t_g \mid L_1 = l])\hat{\pi}_{(l,m)}. \end{aligned} \tag{39}$$

Assuming that t_g is uniformly distributed in the slot in which the packet is generated, we have,

$$E[t_g \mid L_1 = l] = \frac{p\sum_{k=1}^{l} k(1-p)^{k-1}}{1-(1-p)^l} - 0.5, \tag{40}$$

where the first term is the probability that the packet is generated in the kth slot of the first TP which has length l. Furthermore,

$$E[t_s \mid L_1 = l, L_2 = m] \leq l + m, \tag{41}$$

we thus have, from (39),

$$D \leq \sum_{l,m \in \mathcal{Q}_2} (l+m - \frac{p\sum_{k=1}^{l} k(1-p)^{k-1}}{1-(1-p)^l} + 0.5)\hat{\pi}_{(l,m)}$$
$$= \sum_{l,m \in \mathcal{Q}_2} (l+m - \frac{p\sum_{k=1}^{l} k(1-p)^{k-1}}{1-(1-p)^l} + 0.5)\pi_l p_{l,m}. \quad (42)$$

¿From (35,42) we see that the throughput and average delay provided by the dynamic queue protocol are given by the limiting distribution $\{\pi_l\}_{l\in\mathcal{S}}$ and the transition probability $\{p_{l,m}\}_{l,m\in\mathcal{S}}$ of $\{L_i\}_{i=0}^{\infty}$. In general, these two quantities are difficult to obtain even numerically. For simple examples, however, they may be studied analytically as shown in Section V-A. Another special case where the throughput and average delay can be computed numerically is when $p=1$, which we discuss in Section IV-B.3.

B.3 Throughput and Average Delay at $p=1$

At $p=1$, we have $q_i=1$ for any i. It then follows that $\{L_i\}_{i=0}^{\infty}$ is an i.i.d. sequence. The throughput and average delay for $p=1$ are given by

$$U = \frac{M}{\min_{N=1,\cdots,M} E[L_i \mid q_i=1,N]}, \quad (43)$$
$$D \leq 2\min_{N=1,\cdots,M} E[L_i \mid q_i=1,N] - 0.5. \quad (44)$$

As shown in Section III-C, $E[L_i \mid q_i=1,N]$ can be obtained by analyzing the absorbing time of a finite state Markov chain as illustrated in Figure 4. With $q_i=1$ for all i, we can simply the state of this Markov chain to the number j of unprocessed users. The transition probability then becomes

$$r_{j,l} = \begin{cases} C_{\min\{N,j\},\ j-l} & \text{if } 0\leq l\leq j \\ 0 & \text{otherwise} \end{cases}. \quad (45)$$

The initial condition of this Markov chain is given by

$$P[X_0=M]=1. \quad (46)$$

With state 0 defined as the absorbing state, $E[L_i \mid q_i=1,N]$ can be obtained as

$$E[L_i \mid q_i=1,N] = e_M, \quad (47)$$

where e_M is the absorbing time of the Markov chain. With $E[L_i \mid q_i=1,N]$ computed for all possible N, the throughput and an upper bound on the average delay at $p=1$ can be easily obtained from (43,44).

V. Numerical and Simulation Examples

A. A Numerical Example

We first study a simple numerical example with $M = 2$. The channel reception matrix is given by

$$\mathbf{C} = \begin{pmatrix} 1-p_1 & p_1 & 0 \\ 1-p_2 & p_2 & 0 \end{pmatrix}, \tag{48}$$

where $0 < p_1, p_2 \leq 1$. By analyzing the absorbing time of the Markov chain with transition probability given by (8), we have

$$\begin{aligned} E[L_i \mid q_i, N_i = 1] &= 2 + 2(1-p_1)q_i/p_1, \\ E[L_i \mid q_i, N_i = 2] &= 1 + \frac{2}{p_1}q_i + \frac{p_1 - p_2 - p_1p_2}{p_1p_2}q_i^2. \end{aligned} \tag{49}$$

Thus, to minimize $E[L_i]$, we have, for the case of $p_2 < p_1$,

$$N_i = \begin{cases} 2 & \text{if } q_i \leq q^* \\ 1 & \text{if } q_i > q^* \end{cases}, \tag{50}$$

where

$$q^* = \begin{cases} \frac{1}{2} & \text{if } p_2 = \frac{p_1}{1+p_1} \\ \frac{\sqrt{p_1p_2(p_1-p_2)} - p_1p_2}{p_1-p_2-p_1p_2} & \text{otherwise} \end{cases}. \tag{51}$$

However, when $p_2 \geq p_1$, we have $N_i = 2$ for all q_i, *i.e.*, when the MPR capability is sufficiently strong, contention is desirable at any traffic load. In this example, we consider $p_2 < p_1$ which is usually the case.

By taking into account the incoming traffic load $0 < p < 1$, N_i as a function of L_{i-1} can be obtained from (50) as

$$N_i = \begin{cases} 2 & \text{if } 1 \leq L_{i-1} < \underline{l}_2 \\ 1 & \text{if } L_{i-1} \geq \underline{l}_2 \end{cases}, \tag{52}$$

where

$$\underline{l}_2 = \lceil \frac{\ln(1-q^*)}{\ln(1-p)} \rceil. \tag{53}$$

Suppose that $p \geq 1 - \sqrt{1-q^*}$. In this case, we have $\underline{l}_2 \leq 2$. From Theorem 1 we conclude that

$$\mathcal{Q}_1 = \{1\}, \qquad \mathcal{Q}_2 = \{2, 3, \cdots\} \tag{54}$$

contain, respectively, the transient states and positive recurrent states of $\{L_i\}_{i=0}^{\infty}$. It can be shown that the transition probability for states in $\mathcal{Q}_2$ is given by

$$p_{l,m} = \begin{cases} (1-(1-p)^l)^2 p_1^2 + 2(1-p)^l(1-(1-p)^l)p_1 + (1-p)^{2l} & \text{if } m = 2 \\ (1-(1-p)^l)^2(m-1)p_1^2(1-p_1)^{m-2} + 2(1-p)^l(1-(1-p)^l)p_1(1-p_1)^{m-2} & \text{if } m > 2 \end{cases}. \quad (55)$$

The limiting distribution of $\{L_i\}_{i=0}^{\infty}$ is then given by

$$\pi_l = \begin{cases} ap_1^2 + bp_1 + (1-a-b) & \text{if } l = 2 \\ a(l-1)p_1^2(1-p_1)^{l-2} + bp_1(1-p_1)^{l-2} & \text{if } l > 2 \end{cases}, \quad (56)$$

where a and b are, respectively, the probability that both users have packet and that only one user has packet at the beginning of a TP in the steady state. It is difficult to obtain them in close form. However, by estimating them through simulations, we can easily obtain the limiting distribution of $\{L_i\}_{i=0}^{\infty}$. The throughput can then be calculated from (35).

We now consider an example with $p_1 = \frac{3}{4}$ and $p_2 = \frac{1}{2}$. The limiting distribution of $\{L_i\}_{i=0}^{\infty}$ at $p = 0.4$ and $p = 0.9$ are shown in Figure 7-left, where we see that π_l decays exponentially in l, as promised by (56). Comparing the limiting distribution at $p = 0.4$ and that at $p = 0.9$, we see that $E[L_i]$ increases with the incoming traffic load p. The calculated and the simulated throughput of this 2-user system are shown in Figure 7-right as a function of p. In both figures, the theoretical results match perfectly with the simulation results.

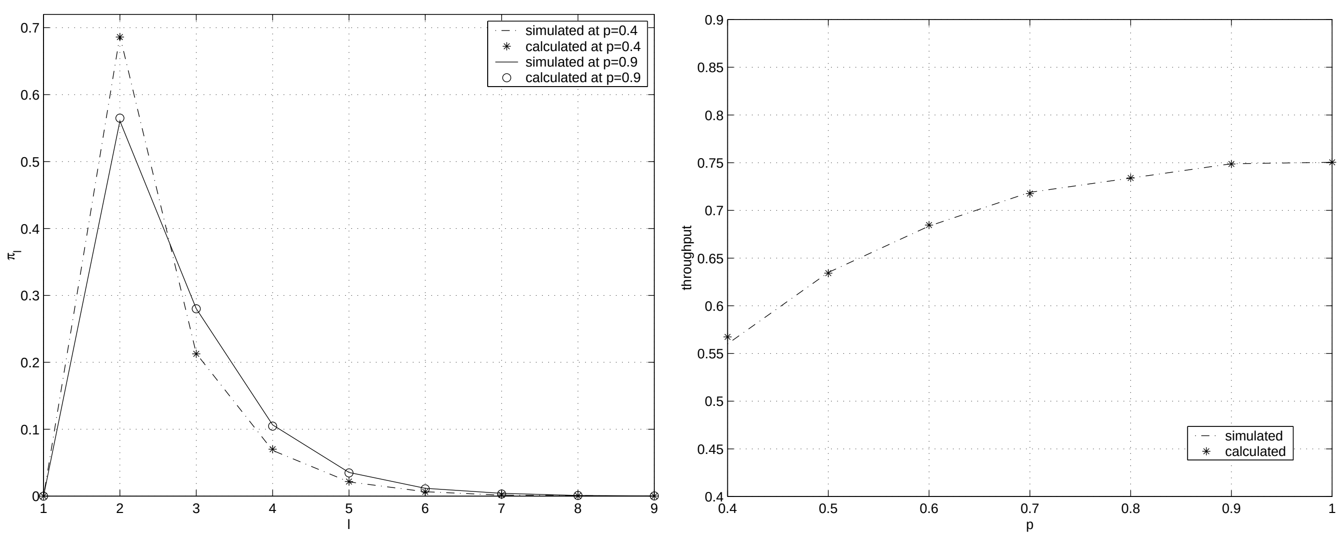

Fig. 7: The limiting distribution and the throughput for the 2-user example.

figure

B. Simulation Examples: MPR via Spread Spectrum

In this example, we consider a CDMA network with M users. Each transmitted packet is spread by a randomly generated code with length P. At the central controller, the spreading code of each transmitted packet is assumed known, and a bank of matched filters are used as the receiver. We assume that each packet contains L_p bits. A block error control code is used which corrects up to t errors in each received packet. We consider a noisy environment where the variance of the additive white Gaussian noise is denoted by σ^2.

We first construct the reception matrix $\mathbf{C}$ for such a network. Under the Gaussian assumption on the multiaccess interference from users with equal power, the bit-error-rate (BER) p_e of a packet received in the presence of $n-1$ interfering packets is given by [18]

$$p_e(n-1) = Q(\sqrt{\frac{3P}{n-1+3P\sigma^2}}). \tag{57}$$

Assuming that errors occur independently in a packet, we then have the packet success probability in the presence of $n-1$ interfering packets as

$$p_s(n-1) = \sum_{i=0}^{t} B(i, L_p, p_e). \tag{58}$$

Under the assumption that each matched filter works independently at the receiver, we have

$$C_{n,k} = B(k, n, p_s(n-1)). \tag{59}$$

B.1 Throughput

In this example, we compare the throughput performance of the dynamic queue protocol with that of the MQSR protocol and the slotted ALOHA with optimal retransmission probability. We considered a network with $M = 10$. The packet length L_p, spreading gain P, and the number of correctable errors in a packet were, respectively, 200, 6, and 2. The noise variance was given by $10\log_{10}\frac{1}{\sigma^2} = 10dB$. The capacity of the MPR channel in such a network is 1.7925, which can be achieved by transmitting $n_0 = 2$ packets in each slot.

We first construct the look-up table that specifies the q_i intervals in which a possible size (from 1 to 10) of access set is optimal. The result is shown in Figure 8. This result demonstrates clearly the trend that the heavier the traffic is (larger q_i), the smaller the access set should be, as intuition suggests.

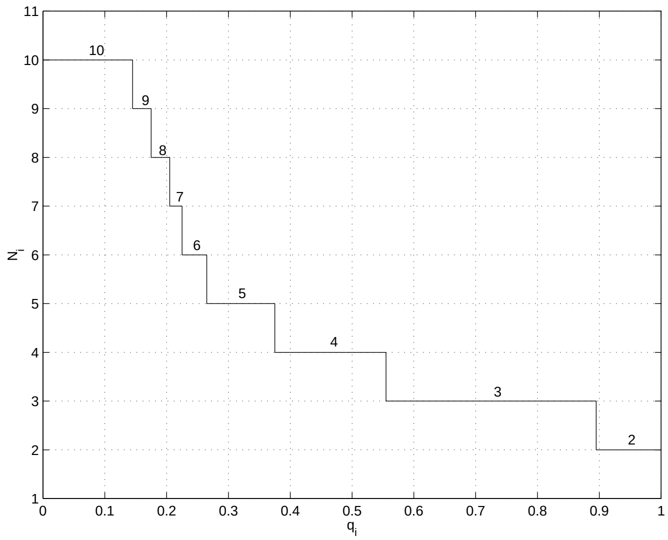

Fig. 8: The optimal size of access set.

figure

Note that the optimal size of access set equals to n_0 which is greater than 1 at the heaviest traffic load ($q_i = 1$), indicating that contention is preferable at any traffic load for this MPR channel.

In Figure 9, the throughput performance of the dynamic queue protocol at different incoming traffic load p is compared to that of the multi-queue service room (MQSR) protocol [29] and the slotted ALOHA with delayed first transmission. Here we intentionally favored the slotted ALOHA by letting it choose the optimal retransmission probability. Comparing the performance of the dynamic queue protocol with that of the slotted ALOHA with optimal retransmission probability, we see a 55% throughput gain at medium and heavy traffic load. Compared to the MQSR protocol which aims to determine the access set for each slot by optimally exploiting all available information, the dynamic queue protocol achieved comparable performance with a much simpler implementation. Note that the throughput provided by the dynamic queue protocol at heavy traffic load approached to the channel capacity 1.7925.

B.2 Average Delay

Here we study the delay performance of the dynamic queue protocol in the CDMA network specified in Section V-B.1. We first study the expected length of a TP in the dynamic queue protocol, which is closely related to the average packet delay. In Figure 10, the expected length of a TP is plotted as a function of q, the probability that a user has a packet to transmit in this TP. Compared to schemes

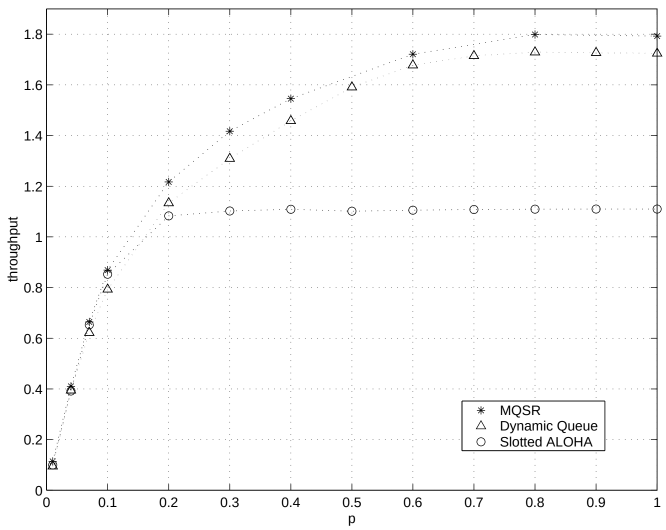

Fig. 9: Throughput comparison.

figure

with fixed size N of access set, the advantage of dynamically changing the size of access set according to the traffic load q is obvious.

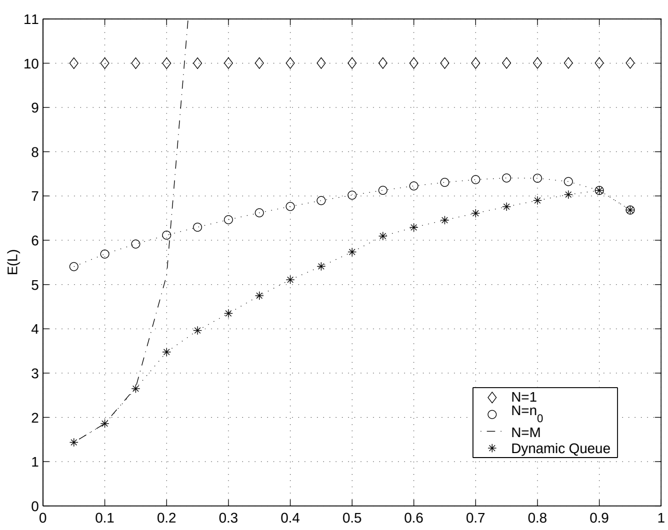

Fig. 10: The expected length of a TP.

figure

Shown in Figure 11 is the average packet delay provided by the dynamic queue protocol as a function of the incoming traffic load p. We consider two cases – fixed order of users and randomized order of users. From Figure 11 we see that in the case of fixed order of users, the average delay for the last user in the queue could be twice as large as that for the first user in the queue at medium and heavy traffic load, while at light traffic load, they were about the same. The reason for this is that at medium and heavy traffic load, $N_i < M$ for most transmission periods. In this case, the first user in the queue always access the channel before the last one. While at light traffic load when we usually have $N_i = M$, all users access the channel simultaneously, resulting in the same average delay for the first and last user in the queue. For the case of randomizing the order of users, the average packet delay for a user was about the average of the delay for the first user and the delay for the last user in the case of fixed order of users.

¿From Figure 11 we can also see that the average length of a TP $E[L]$ could be a good estimate of the packet delay for the first user in the queue. At medium traffic load ($0.1 \leq p \leq 0.6$), the average delay for the first user in the queue was slightly smaller than $E[L]$, while at heavy traffic load ($p > 0.6$), it was the other way around. The reason for this is that when the traffic is heavy, with high probability, a user will generate a packet within the first several slots in a TP. Even for the first user in the queue, it has to wait for almost a whole TP before this packet can be transmitted. For the last user in the queue, its average delay will approach the length of two transmission periods at heavy traffic load, as confirmed by Figure 11.

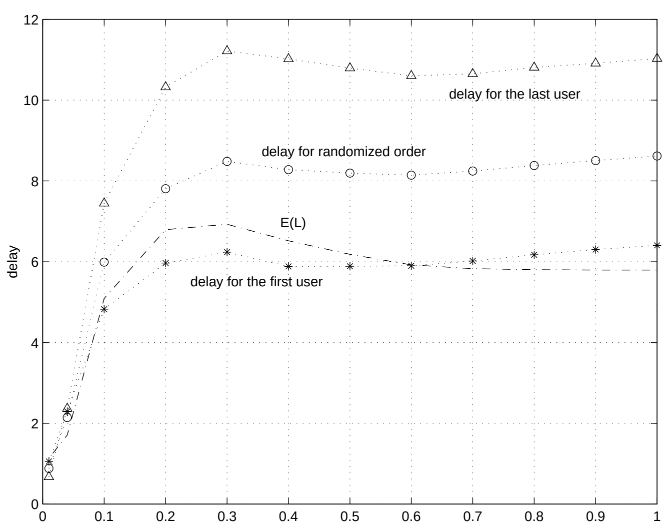

Fig. 11: Average packet delay.

figure

B.3 Normalized Throughput

While the techniques of spread spectrum and error control strengthen the channel reception capability, they consume bandwidth. In this example, we study the normalized throughput of the dynamic queue protocol, where we define the normalized throughput as the average number of information bits successfully transmitted per second per Hertz [23]. We assume here a BPSK modulation. Given the network throughput U, spreading gain P, packet length L_p, coding rate r_c, and symbol duration T_s, the average number of successfully transmitted information bits per slot is $L_p r_c U$; the duration of each time slot is $L_p T_s$ and the bandwidth $\frac{P}{T_s}$. Hence, the normalized throughput U_n is given by

$$U_n = \frac{L_p r_c U}{L_p T_s \frac{P}{T_s}} = \frac{r_c}{P} U. \tag{60}$$

As shown in [15], the maximum coding rate r_c can be computed from the number t of correctable errors as follows.

$$\begin{aligned} \alpha &= \frac{2t+1}{L_p} \\ r_c &= 1 + \alpha \log_2(\alpha) + (1-\alpha)\log_2(1-\alpha). \end{aligned}$$

We compare the normalized throughput of the dynamic queue protocol with that of the MQSR protocol and the slotted ALOHA with optimal retransmission probability. We choose $p = 1$ for the reason that all three protocols yield maximum throughput at this heaviest traffic load. The network parameters were chosen as $M = 200$, $L_p = 1000$, $P = 10$, and $\sigma^2 = 0$. The normalized throughput of the dynamic queue, the MQSR, and the slotted ALOHA with optimal retransmission probability at $p = 1$ was theoretically calculated and plotted in Figure 12 as t, the number of correctable errors within one packet, varies from 0 to 150. From Figure 12, we again observe that the dynamic queue protocol performed comparably to the optimal MQSR protocol and significantly better than the slotted ALOHA with optimal retransmission probability. Note that the MQSR protocol achieves the channel capacity at $p = 1$ which has been shown theoretically in [29]. A comparable performance to it implies that the throughput provided by the dynamic queue protocol approaches to the channel capacity at heavy traffic load. Furthermore, Figure 12 shows that to achieve the best bandwidth efficiency, we should choose a block error control code which corrects up to $t = 30$ errors out of a packet with 1000 bits for the dynamic queue and the MQSR protocol. For the slotted ALOHA with optimal retransmission probability, however, we should choose $t = 60$. A block error control code with stronger correction capability is, in general, more difficult to design.

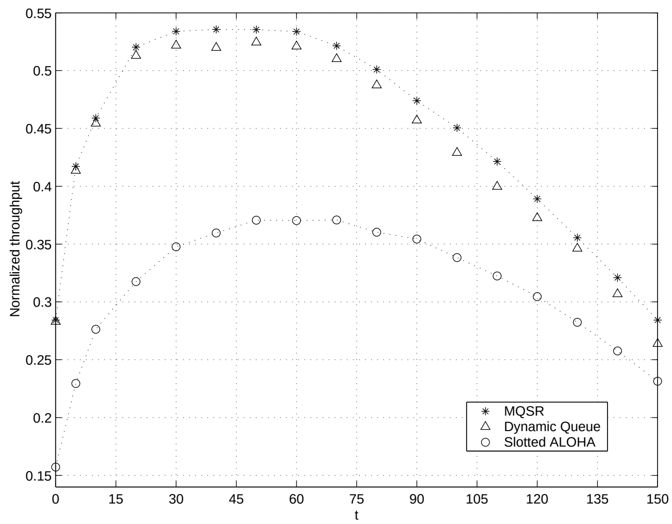

Fig. 12: Normalized throughput at $p = 1$.

figure

VI. Conclusion

In this paper, we propose the dynamic queue protocol for multiaccess networks with MPR capability. According to the traffic load and the channel MPR capability, this protocol adaptively controls the number of users who gain access to the channel in the same slot. As a consequence, unnecessary empty slots at light traffic and excessive collision events at heavy traffic are avoided simultaneously, leading to efficient channel utilization at any incoming traffic load. Furthermore, the proposed protocol is particularly attractive in its simple implementation.

Appendix A

On the Markovian Property of $\{L_i\}_{i=0}^{\infty}$

Given the channel reception matrix $\mathbf{C}$ and the traffic load p, the distribution of L_i is completely determined by q_i and N_i (see Figure 4), which in turn, are determined by L_{i-1} through (6) and (15). We thus have

$$P[L_i = l_i \mid L_{i-1} = l_{i-1}, \cdots, L_0 = l_0] = P[L_i = l_i \mid L_{i-1} = l_{i-1}] \triangleq p_{l_{i-1}, l_i}, \tag{61}$$

i.e., $\{L_i\}_{i=0}^{\infty}$ form a Markov chain. Since the transition probability $\{p_{l,m}\}$ is independent of the TP index i, this Markov chain is homogeneous. We consider here an initial condition L_0 which can take any positive integers, resulting in an infinite state space $\mathcal{S} = \mathcal{Z}^+$.

□□□

Appendix B

Proof of Property 1

We start with *P1.2*, which follows directly from the following two facts.

1. If $N_i = \underline{n}_j$ $(j = 1, \cdots, J)$, the minimum value that L_i can take is $\lceil \frac{M}{\underline{n}_j} \rceil$.

2. Based on $f_{\mathbf{C},p}(\cdot)$ being monotonically decreasing, when $L_{i-1} = l \geq \underline{l}_j$, we have $N_i \leq \underline{n}_j$.

Consider $L_{i-1} = l \geq \lceil \frac{M}{\underline{n}_w} \rceil \geq \underline{l}_w$. By the second statement above, we have $N_i \leq \underline{n}_w$. Then by the first statement, we have

$$L_i \geq \lceil \frac{M}{N_i} \rceil \geq \lceil \frac{M}{\underline{n}_w} \rceil, \tag{62}$$

i.e., starting with a state no smaller than $\lceil \frac{M}{\underline{n}_w} \rceil$, the chain can never hit a state smaller than $\lceil \frac{M}{\underline{n}_w} \rceil$. This proves *P1.2*.

We now consider *P1.1*. Similar to the second statement above, when $L_{i-1} = l < \underline{l}_{j+1}$, we have $N_i \geq \underline{n}_j$. Hence,

$$\lceil \frac{M}{\underline{n}_j} \rceil \geq \lceil \frac{M}{N_i} \rceil. \tag{63}$$

For $m \geq \lceil \frac{M}{\underline{n}_j} \rceil$, we have $m \geq \lceil \frac{M}{N_i} \rceil$. When equality holds, by considering the event that no packet has been generated in the $(i-1)$ TP, we have

$$p_{l,m} \triangleq P[L_i = m \mid L_{i-1} = l] \geq (1 - q_i)^M > 0, \tag{64}$$

where

$$0 < q_i = 1 - (1-p)^l < 1. \tag{65}$$

When $m > \lceil \frac{M}{N_i} \rceil$, we consider the event that only the first user in the queue has generated a packet in the $(i-1)$th TP. Thus,

$$p_{l,m} \geq q_i (1 - q_i)^{M-1} C_{1,0}^{m - \lceil \frac{M-1}{N_i} \rceil} C_{1,1} > 0, \tag{66}$$

which completes the proof of *P1.1*.

□□□

References

[1] N. Abramson. "The Aloha System - Another Alternative for Computer Communications". In *Proc. Fall Joint Comput. Conf., AFIPS Conf.*, page 37, 1970.

[2] N. Abramson. "The Throughput of Packet Broadcasting Channels". *IEEE Trans. Comm*, COM-25(1):117–128, January 1977.

[3] D.E. Ayyildiz and H. Delic. "Adaptive Random Access Algorithm with Improved Delay Performance". *Int. J. Commun. Syst.*, 14:531–539, 2001.

[4] J.Q. Bao and L. Tong. "A Performance Comparison Between Ad Hoc and Centrally Controlled CDMA Wireless LANs". *To appear in IEEE Trans. Wireless Communications.*, 2002.

[5] J.I. Capetanakis. "Generalized TDMA: The Multi-Accessing Tree Protocol". *IEEE Trans. Communications*, 27(10):1476–1484, Oct. 1979.

[6] J.I. Capetanakis. "Tree Algorithms for Packet Broadcast Channels.". *IEEE Trans. Information Theory*, 25(5):505–515, Sept. 1979.

[7] I. Chlamtac and A. Farago. "An Optimal Channel Access Protocol with Multiple Reception Capacity". *IEEE Trans. Computers*, 43(4):480–484, April 1994.

[8] G. del Angel and T.L. Fine. "Randomized Power Control Strategies for Optimization of Multiple Access Radio Systems". In *Proc. 38th Allerton Conference on Communication, Control and Computing*, October 2000.

[9] R. Gallager. "Conflict Resolution in Random Access Broadcast Networks". In *Proc. of AFOSR Workshop on Comm. Theory and Appl.*, pages 74–76, Sept. 1978.

[10] S. Ghez, S. Verdú, and S.C. Schwartz. "Stability Properties of Slotted Aloha with Multipacket Reception Capability" . *IEEE Trans. Automat. Contr.*, 33(7):640–649, July 1988.

[11] S. Ghez, S. Verdú, and S.C. Schwartz. "Optimal Decentralized Control in the Random Access Multipacket Channel". *IEEE Trans. Automat. Contr.*, 34(11):1153–1163, Nov. 1989.

[12] I.M.I. Habbab and *et al.* "ALOHA with Capture over Slow and Fast Fading Radio Channels with Coding and Diversity". *IEEE Journal on Select. Areas in Comm.*, 7:79–88, January 1989.

[13] M.G. Hluchyj. "Multiple Access Window Protocol: Analysis for Large Finite Populations". In *Proc. IEEE Conf. on Decision and Control*, pages 589–595, New York, NY, 1982.

[14] M.G. Hluchyj and R.G. Gallager. "Multiaccess of A Slotted Channel by Finitely Many Users". In *Proc. Nat. Telecomm. Conf.*, pages D4.2.1–D4.2.7, New Orleans, LA., Aug. 1981.

[15] R.K. Morrow Jr. and J.S. Lehnert. "Packet Throughput in Slotted ALOHA DS/SSMA Radio Systems with Random Signature Sequences". *IEEE Trans. Comm.*, 40(7):1223–1230, July 1992.

[16] S. Kim and J. Yeo. "Optimal Scheduling in CDMA Pakcet Radio Networks". *Computers and Operations Research*, 25:219–227, March 1998.

[17] L. Kleinrock and Y. Yemini. "An Optimal Adaptive Scheme for Multiple Access Broadcast Communication". In *Proc. International Conference on Communications*, pages 7.2.1–7.2.5, June 1978.

[18] J. Lehnert and M. Pursley. "Error Probabilities for Binary Direct-Sequence Spread-Spectrum Communications with Random Signature Sequences". *IEEE Trans. on Communications*, COM-35(1):87–98, January 1987.

[19] D.F. Lyons and P. Papantoni-Kazakos. "A Window Random Access Algorithm for Environments with Capture". *IEEE Trans. on Communications*, 37(7):766–770, July 1989.

[20] J.J. Metzner. "On Improving Utilization in ALOHA Networks". *IEEE Trans. Communications*, 24:447–448, Apr. 1976.

[21] A.G. Pakes. "some conditions for ergodicity and recurrence of markov chains". *Operation Research*, 17:1058–1061, 1969.

[22] M. Paterakis and P. Papantoni-Kazakos. "A Simple Window Random Access Algorithm with Advantageous Properties". *IEEE Trans. Information Theory*, 35(5):1124–1130, September 1989.

[23] A. Polydors and J. Sylvester. "Slotted Random Access Spread-Spectrum Networks: An Analytical Framework". *IEEE Journal on Selected Areas in Communications*, SAC-5:989–1002, July 1987.

[24] L.G. Roberts. "Aloha Packet System with and without Slots and Capture". In *(ASS Note 8). Stanford, CA: Stanford Research Institute, Advanced Research Projects Agency, Network Information Center.*, 1972.

[25] F. Shad, T.D. Todd, V. Kezys, and J. Litva. "Dynamic Slot Allocation (DSA) in Indoor SDMA/TDMA Using a Smart Antenna Basestation". *IEEE/ACM Trans. on Networking*, 9(1):69–81, February 2001.

[26] M. Sidi and I. Cidon. "Splitting Protocols in Presence of Capture". *IEEE Trans. Info. Theory*, IT-31:295–301, March 1985.

[27] C. Vanderplas and J.P.M. Linnartz. "Stability of Mobile Slotted ALOHA Network with Rayleigh Fading, Shadowing, and Near-far Effect". *IEEE Trans. Veh. Tech.*, 39:359–366, November 1990.

[28] B. Yucel and H. Delic. "Mobile Radio Window Random Access Algorithm with Diversity". *IEEE Trans. Vehicular Technology*, 49(6):2060–2070, November 2000.

[29] Q. Zhao and L. Tong. "A Multi-Queue Service Room MAC Protocol for Wireless Networks with Multipacket Reception". *IEEE/ACM Transactions on Networking (http://people.ece.cornell.edu/ltong)*, February 2003.

[30] M. Zorzi. "Mobile Radio Slotted ALOHA with Capture and Diversity". *Wireless Networks*, 1:227–239, May 1995.

[31] M. Zorzi and R.R. Rao. "Capture and Retransmission Control in Mobile Radio". *IEEE Journal on Select. Areas in Comm.*, 12:1289–1298, Oct. 1994.